

**PETA INTERAKTIF PENCARIAN JALUR TERPENDEK
DENGAN MENGGUNAKAN ALGORITMA SEMUT
UNTUK PENJEMPUTAN BARANG
(Studi Kasus PT.TIKI Pekanbaru)**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

oleh :

M. AIDIL ADHA

10351022924



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM
PEKANBARU
2010**

**PETA INTERAKTIF PENCARIAN JALUR TERPENDEK
DENGAN MENGGUNAKAN ALGORITMA SEMUT UNTUK
PENJEMPUTAN BARANG
(Studi Kasus PT.TIKI Pekanbaru)**

M. AIDIL ADHA

10351022924

Tanggal Sidang: 29 Januari 2010

Periode Wisuda: Februari 2010

Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

Permasalahan yang dihadapi oleh PT. TIKI Pekanbaru adalah sering terlambatnya penjemputan barang. Masalah lain adalah tidak terpetanya sub TIKI yang tersebar di Kota Pekanbaru. Pada tugas akhir ini dirancang dan dibangun sebuah peta interaktif pencarian jalur terpendek menggunakan algoritma semut. Parameter yang digunakan adalah sub TIKI, nama jalan dan jarak jalan. Masukan pada algoritma semut adalah *Pheromon* awal, Jumlah Semut, Faktor *update* lokal, Faktor *update* global dan jumlah siklus.

Dari pengujian yang dilakukan, sistem akan memberikan gambaran peta perjalanan dan dapat merekomendasikan jarak terpendek untuk penjemputan barang.

Kata kunci : Algoritma semut, Peta interaktif, Pencarian jalur terpendek.

**INTERACTIVE MAP FOR SEARCH OF SHORT PATH
GOODS FETCHING USING ANT ALGORITHM
(Case Study PT.TIKI Pekanbaru)**

M. AIDIL ADHA

10351022924

Date of Final Exam : Januari 29th, 2010

Graduation Cremony Priod : Februari 2010

Engineering Departement of Informatic Technology

Faculty of Sciences and Technology

State Islamic University of Sultan Syarif Kasim Riau

ABSTRACT

The problems that PT TIKI Pekanbaru faced is often late of goods fetching. Other problems is unmapped of sub TIKI which spread over in Pekanbaru. this final project will be design and develop a interactive map for search of short path using the ant algorithm. The parameters that used are sub TIKI, street name, and distance of road. Input for the ant algorithm is initial pheromone, the number of ants, local update factor, global update factor and the number of cycle.

The result test which have been done, the system will give the picture map journey and can be recommendation short path for pick up goods.

Keywords: ant algorithms, interactive maps, shortest searching path

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	iv
LEMBAR PERNYATAAN	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK.....	vii
ABSTRACT	viii
KATA PENNGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN.....	xvi
BAB I PENDAHULUAN	I-1
1.1 Latar Belakang	I-1
1.2 Rumusan Masalah	I-2
1.3 Tujuan	I-2
1.4 Batasan Masalah.....	I-2
1.5 Sistematika Penulisan	I-3
BAB II LANDASAN TEORI.....	II-1
2.1 Pengertian Peta.....	II-1
2.2 Konsep Jalur Terpendek.....	II-3
2.3 Algoritma	II-5
2.4 Teknik Pencariah Heuristik.....	II-6

2.4.1 Pencarian Buta (<i>Blind Search</i> atau <i>Un-Informed Search</i>	II-6
2.4.2 Pencarian Terbimbing (<i>Heuristic Search</i> atau <i>Informed Search</i>	II-6
2.5 Algoritma Ant Colony Optimization (ACO)	II-12
2.5.1 Sejarah Algoritma ACO.....	II-12
2.5.2 Teori Algoritma ACO	II-13
2.5.3 Konsep Dasar Algoritma Semut	II-14
2.6 Penerapan Algoritma Semut	II-19
 BAB III METODOLOGI PENELITIAN.....	III-1
3.1 Identifikasi Masalah	III-2
3.2 Perumusan Masalah	III-2
3.3 Pengumpulan Data	III-2
3.4 Pemilihan Metode	III-2
3.5 Analisis Sistem.....	III-3
3.5.1 Sistem Berjalan	III-3
3.5.2 Sistem Informasi yang akan dibangun	III-3
3.6 Desain dan Perancangan Program.....	III-3
3.7 Pengujian dan Penerapan	III-3
3.8 Kesimpulan dan Saran.....	III-4
 BAB IV ANALISIS DAN PERANCANGAN	IV-1
4.1 Analisis Sistem.....	IV-1
4.1.1 Analisis sistem yang sedang berjalan.....	IV-1
4.1.2 Analisis sistem baru	IV-2
4.2 Analisis Pengembangan Algoritma Semut Untuk Mencari Nilai Optimal	IV-2
4.3 Analisis Data Sistem	IV-8

4.3.1 Analisis Input	IV-8
4.3.2 Analisis Proses	IV-8
4.3.3 Analisis Output	IV-9
4.4 Contoh Penerapan Algoritma Semut Dalam Proses	
Pencarian Jalur Terpendek	IV-9
4.5 Perancangan Perangkat Lunak	IV-16
4.5.1 Lingkungan Perancangan	IV-16
4.5.2 Perancangan Antar Muka	IV-16
BAB V IMPLEMENTASI DAN PENGUJIAN	V-1
5.1 Implementasi Sistem	V-1
5.1.1 Lingkungan Implementasi	V-1
5.1.2 Implementasi Peta Interaktif	
Pencarian Jalur Terpendek Dengan Menggunakan Algoritma	
Semut	V-2
5.2 Pengujian Sistem	V-4
5.2.1 Lingkungan Pengujian Sistem	V-4
5.2.2 Pengujian	V-4
5.2.3 Hasil Pengujian	V-4
5.2.4 Kesimpulan Pengujian	V-5
BAB VI PENUTUP	VI-1
6.1 Kesimpulan	VI-1
6.2 Saran	VI-1
DAFTAR PUSTAKA	xvii
LAMPIRAN	

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada awal diciptakan, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan jaman, maka peran komputer semakin mendominasi kehidupan. Lebih dari itu, komputer diharapkan dapat digunakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia baik dalam bidang pendidikan, kesehatan, industri, dan kehidupan sehari-hari sehingga peran komputer dan manusia akan saling melengkapi. Beberapa hal yang menjadi kekurangan manusia diharapkan dapat digantikan oleh komputer. Begitu juga dengan komputer yang tak akan berguna tanpa sentuhan manusia.

PT.TIKI (*Titipan Kilat*) adalah perusahaan yang bergerak dalam bidang jasa pengiriman barang, barang atau paket yang akan dikirim terlebih dahulu harus dijemput dari sub-sub TIKI yang ada di Kota Pekanbaru. Sub TIKI adalah mitra usaha yang bekerja sama dengan PT.TIKI dalam penyedia layanan pengiriman barang. Mitra usaha TIKI adalah Wartel (*Warung Telkom*) dan Biro Travel (*pemesanan tiket perjalanan*) yang tersebar di Kota Pekanbaru. Setelah barang dijemput dari sub TIKI barulah barang atau paket tersebut dikirim ke tujuan. Dalam penjemputan barang ke sub TIKI tersebut sering kali dihadapi tidak terpetanya sub TIKI yang tersebar di Kota Pekanbaru, sehingga kurir yang menjemput barang tersebut harus bolak-balik menjemput barang dengan menggunakan jalan yang berputar atau jalan yang jauh. Permasalahan lain yang dihadapi ialah, komplain dari pihak sub TIKI karena barang yang sudah menumpuk terlambat untuk dijemput, sehingga terjadi komplain oleh konsumen yang mengirimkan barang lewat jasa TIKI.

Dari kasus PT.TIKI diatas dapat disimpulkan bahwa bagaimana membuat suatu sistem yang dapat memetakan objek tujuan penjemputan barang sekaligus

mencari jalur terpendek dalam penjemputan barang tersebut. Untuk menjawab permasalahan tersebut dapat dirancang suatu aplikasi menggunakan peta interaktif untuk memetakan objek penjemputan barang dan algoritma untuk memperoleh jalur terpendek dalam penjemputan barang.

Saat ini penerapan Algoritma komputer dalam menyelesaikan persoalan-persoalan semakin banyak digunakan di dalam dunia nyata. Contohnya adalah *Traveling Salesman Problem* (TSP). “Algoritma” dapat di definisikan sebagai urutan langkah-langkah untuk memecahkan permasalahan (*wikipedia.org*).

Algoritma Semut merupakan satu diantara sekian banyak algoritma yang dapat digunakan untuk menyelesaikan persoalan-persoalan klasik seperti *traveling salesman problem* atau *shortest path problem*, yang pada prinsipnya mencari jalur terpendek dari semua tempat yang harus ditempuh.

1.2 Rumusan Masalah

Bagaimana memanfaatkan algoritma semut untuk pencarian jalur terpendek untuk penjemputan barang pada PT.TIKI dan ditampilkan dalam bentuk Peta Interaktif.

1.3 Tujuan

Tujuan yang ingin dicapai dalam penyusunan tugas akhir ini adalah :

1. Menerapkan teknologi Peta Interaktif pencarian jalur terpendek.
2. Menerapkan Algoritma Semut dalam pencarian jalur terpendek untuk penjemputan barang dari sub-sub TIKI.

1.4 Batasan Masalah

Batasan masalah dalam penyusunan tugas akhir ini adalah :

1. Hanya membahas jarak tempuh.
2. Lokasi awal penjemputan tetap dan tujuan penjemputan dapat diubah sesuai dengan keinginan.
3. Jalan yang hanya bisa dilewati oleh mobil penjemput.
4. Wilayah penjemputan meliputi kecamatan Pekanbaru Kota, Senapelan, Sukajadi, Payung Sekaki.

5. Tidak meliputi lampu lalu lintas (Traffic Light) dan penutupan jalan.

1.5 Sistematika Penulisan

Sistematika penulisan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, tujuan, serta sistematika penulisan laporan tugas akhir.

BAB II LANDASAN TEORI

Bab ini membahas teori-teori yang berhubungan dengan topik penelitian yaitu Peta Interaktif dan Algoritma Semut.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang metodologi yang digunakan dalam penelitian dan pengembangan perangkat lunak.

BAB IV ANALISIS DAN PERANCANGAN

Bab ini membahas tentang hasil analisis, deskripsi sistem, fungsi produk, karakteristik pengguna, deskripsi umum kebutuhan, deskripsi perancangan rinci dan perancangan antar muka sistem informasi.

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas implementasi dan pengujian yang dilakukan terhadap aplikasi Peta Interaktif pencarian jalur terpendek dengan menggunakan Algoritma Semut untuk penjemputan barang.

BAB VI PENUTUP

Bab ini berisi kesimpulan yang dihasilkan dari pembahasan tentang aplikasi Peta Interaktif pencarian jalur terpendek dengan menggunakan Algoritma Semut untuk penjemputan barang dan beberapa saran sebagai hasil akhir dari penelitian yang telah dilakukan.

BAB II

LANDASAN TEORI

2.1 Pengertian Peta

Peta adalah gambar atau lukisan keseluruhan atau pun sebagian permukaan bumi baik laut maupun darat. tetapi pada dasarnya peta adalah bayangan/gambaran yang diperkecil dari sebagian besar atau sebagian kecil permukaan bumi pada bidang datar dengan skala dan sistem proyeksi tertentu (*Wongsotjitro, 1980*).

2.1.1 Jenis- jenis Peta

Peta dapat diklasifikasi menjadi dua 2 jenis, yakni :

1. Peta Umum

Peta yang menampilkan bentuk fisik permukaan bumi suatu wilayah. Contohnya adalah Peta jalan dan gedung

2. Peta Khusus

Peta khusus adalah peta yang menampakkan suatu keadaan atau kondisi khusus suatu daerah tertentu atau keseluruhan daerah bumi. Contohnya adalah peta persebaran hasil tambang, peta curah hujan, peta pertanian perkebunan, peta iklim, dan lain sebagainya.

2.1.2 Karakteristik Peta

Peta memiliki beberapa karakteristik sebagai berikut :

1. Gambar disajikan pada bidang datar dalam bentuk dua dimensi (hasil transformasi matematik).
2. Merupakan bentuk reduksi dari keadaan sebenarnya.
3. Dalam penyajiannya mengalami suatu proses generalisasi, sehingga tidak semua informasi dapat disajikan.

4. Merupakan suatu bentuk penegasan (*enhancement*) dari unsur yang terdapat di permukaan bumi (misal : kontur).

2.1.3 Fungsi Peta

Peta mempunyai beberapa fungsi, yaitu :

1. Memperlihatkan posisi atau lokasi relatif dari suatu tempat.
2. Memperlihatkan ukuran dalam pengertian jarak dan arah.
3. Memperlihatkan bentuk atau unsur yang terdapat di permukaan bumi.
4. Menghimpun serta menselektif data permukaan bumi.

2.1.4 Bentuk Penyajian Peta

Berdasarkan penyajiannya, peta dapat dikelompokkan sebagai berikut :

1. Peta Garis (Peta Vektor)

Kenampakan permukaan bumi pada peta disajikan oleh garis (baik hitam putih maupun berwarna) dan area yang dilengkapi dengan teks sebagai tambahan informasi. Unsur yang terdapat di permukaan bumi disajikan dengan simbol atau batas. Dengan kata lain, peta garis adalah peta yang memiliki jarak dan arah.

2. Peta Citra (Peta Foto)

Kenampakan permukaan bumi disajikan dalam bentuk citra (sekumpulan informasi yang berasal dari sensor, perolehan tidak secara kontak langsung dengan obyek permukaan bumi ditempat pengamatan). Bayangan permukaan bumi dapat diperoleh melalui foto udara, radar serta sensor *airborne* lainnya dan citra satelit.

2.1.5 Pembagian Peta

1. Peta Luas

Peta luas adalah peta yang menggambarkan suatu daerah yang luas seperti peta dunia, peta daerah amerika utara, peta benua, peta samudera, peta kutub utara dan kutub selatan, dsb.

2. Peta Sempit

Peta sempit adalah peta yang hanya menampilkan sebagian kecil suatu area. Contoh peta sempit yaitu peta desa atau pedesaan, peta kota atau perkotaan, peta gorong-gorong kampung, peta gedung, denah rumah, dan lain sebagainya.

2.1.6 Bentuk Lain Dari Peta

1. Atlas

Atlas adalah gabungan dari beberapa peta yang dikumpulkan dalam sebuah buku yang memiliki judul atlas serta jenis-jenis atlas yang ada di buku tersebut.

2. Globe

Globe atau Bola Dunia adalah suatu bentuk tiruan bola bumi yang dibuat dalam skala yang kecil untuk dapat lebih memahami bentuk asli planet bumi.

2.1.7 Jenis Skala Pada Peta

Skala peta adalah perbandingan jarak di peta dengan jarak sesungguhnya dengan satuan atau teknik tertentu.

1. Skala Angka / Skala Pecahan

Contohnya seperti 1 : 1000 yang berarti 1 cm di peta sama dengan 1000 cm jarak aslinya di dunia nyata.

2. Skala Satuan

Misalnya seperti 1 inchi to 5 miles dengan arti 1 inch di peta adalah sama dengan 5 mil pada jarak sebenarnya.

3. Skala Garis

Skala garis menampilkan suatu garis dengan beberapa satuan jarak yang menyatakan suatu jarak pada tiap satuan jarak yang ada.

2.2 Konsep Jalur Terpendek

Pencarian jalur terpendek melibatkan proses pencarian jalur, dengan biaya dan hambatan yang terkecil.

Pada umumnya, setiap jaringan jalan memiliki aturan tertentu yang harus di sepakati oleh para pengguna. Antara lain: (Prahasta05)

1. *Travel cost*: biaya rata-rata dalam melintasi sebuah *link*, di modelkan dalam satuan, jarak, waktu, biaya, atau satuan lainnya.
2. *One-way streets* : suatu jalan yang hanya dapat dilalui satu arah .
3. *Turns* : tidak boleh berbelok atau memutar – *left, right, straight, u-turn* pada suatu perpotongan jalan tidak boleh berbelok, atau boleh berbelok dengan resiko bobot biaya yang lebih besar.
4. *Over-and underpasses* : sebuah jalan yang terletak di atas (jembatan/*overpass*) atau di bawah (*underpass*) jalan yang lain dimana pengguna tidak dapat berbelok atau memutar ke arahnya (berjalan di atasnya) secara langsung .
5. *Close streets* : jalan yang baru saja tutup (karena sebab-sebab tertentu) untuk lalu lintas umum, atau tipe jalan tertentu yang tidak memenuhi persyaratan

2.2.1 Graph

Graph secara umum bisa sidefinisikan sebagai kumpulan titik (*nodes* atau *vertices*) dan garis (*edges* atau *arcs*). (Santosa01)

Karena garis selalu diawali pada suatu titik dan diakhiri pada titik yang lain, maka garis bisa dituliskan sebagai pasangan antara dua titik. (Santosa01)

Graph berarah (*directed graph* atau *digraph*) adalah merupakan bentuk yang lebih khusus dari *graph*, karena ke dalam *graph* berarah terkandung suatu aliran (*flow*), misalnya aliran beban, dari suatu titik ke titik yang lainnya, biasa disajikan menggunakan anak panah. (Santosa01)

Indegree, yaitu banyaknya anak panah yang masuk ke dalam sebuah titik (*nodes* atau *vertices*), *outdegree* adalah banyaknya anak panah yang meninggalkan suatu titik (*nodes* atau *vertices*). (santosa01)

Terminologi yang dikenal dalam *graph*

1. *Graph* adalah kumpulan *nodes* atau *nodes* dan *edges*.

2. *Graph* berarah merupakan bentuk yang lebih khusus dari *graph*, ke dalam *graph* berarah terkandung suatu aliran (*flow*).
3. *Vertex* atau *nodes* adalah titik-titik yang terdapat pada *graph*.
4. *Edges* adalah garis yang diawali oleh titik dan di akhiri oleh titik yang lain.
5. Lintasan atau *path* adalah kumpulan titik-titik dengan panjang n dan terdiri dari $n+1$ titik.

Penyajian *graph* berarah menggunakan larik (matrik) cukup banyak variasinya tetapi yang paling sering digunakan adalah matrik jalur dan matrik tetangga. Penyajian *graph* seringkali tidak cukup hanya dengan menggunakan matrik tetangga atau matrik jalur. Karna pada persoalan *tarvelling saleman problem* sering harus dihitung besarnya biaya yang harus dikeluarkan untuk berjalan dari satu kota ke kota yang lain, dengan pikiran bahwa biayanya harus minimal.

Matrik beban pada dasarnya hampir sama dengan matrik tetangga, yaitu sebuah matrik berukuran $n \times n$ (dengan n adalah banyaknya titik) dan didefinisikan sebagai berikut: (Santosa01)

$$W_{ij} = \begin{cases} \text{Beban} & \text{jika ada jalur antara } v_i \text{ dengan } v_j \\ 0 & \text{jika tidak ada jalur antara } v_i \text{ dengan } v_j \end{cases}$$

Beban adalah suatu besaran yang menunjukkan biaya yang harus dikeluarkan (besarnya beban yang harus ditanggung) jika seseorang ingin bepergian dari kota v_i ke kota v_j . (Santosa01)

2.3 Algoritma

Algoritma adalah metode perisis yang dapat digunakan komputer untuk menyelesaikan masalah. Algoritma disusun dari sekumpulan langkah berhingga, masing-masing langkah mungkin memerlukan satu operasi atau lebih. Algoritma umumnya dirancang untuk menyelesaikan suatu masalah spesifik dan dengan usaha yang paling minimal.

Ciri-ciri algoritma antara lain:

1. *Input* : terdapat nol atau lebih masukan yang diberikan secara eksternal.
2. *Output* : sedikitnya terdapat satu keluaran yang dihasilkan.
3. *Definite* : harus secara sempurna menyatakan apa yang dilakukan .
4. *Efective* : setiap intruksi harus dapat dilakukan secara menggunakan pensil dan kertas dalam sejumlah waktu yang berhingga.
5. *Terminate* : harus berhenti setelah sejumlah terbatas operasi.

2.4 Teknik Pencarian Heuristik

Pada dasarnya ada 2 teknik pencarian dan pelacakan yang digunakan, (Kusumadewi, 2003), yaitu:

2.4.1 Pencarian Buta (*Blind Search* atau *Un-Informed Search*)

Terdapat enam metode pencarian buta, yaitu: pencarian melebar pertama (*Breadth First Search*), pencarian mendalam pertama (*Depth First Search*), *Uniform Cost Search* (UCS), *Depth-Limited Search* (DLS), *Iterative-Deepening Search* (IDS), dan *Bi-Directional Search* (BDS) (Suyanto, 2007).

Problem pada *blind search* (*Brute Force*, *Systematic Search*, dan *Un-Informed Search*) merupakan *combinatorial explosion*. Inilah yang menyebabkan *blind search* sering disebut sebagai *exhaustive search* (pencarian solusi yang melelahkan) (Gunawan, 2005).

2.4.2 Pencarian Terbimbing (*Heuristic Search* atau *Informed Search*)

Pencarian *heuristic* adalah metoda pencarian yang berusaha memperbaiki efesiensi proses pencarian, yang mungkin dilakukan dengan cara mengorbankan ketidaklengkapan. *Heuristic* berarti “saya telah menemukan” (Suyoto, 2004).

Penyelesaian masalah dengan menggunakan *heuristic* disebabkan atas dua keadaan dasar (George F Luger, 1998) (Yusra, 2007), yaitu:

1. Masalah yang tidak memiliki solusi pasti (*Unexact Solution*)

Dalam hal ini dikarenakan ambigu yang melekat pada data yang tersedia.

2. Masalah yang memiliki solusi pasti (*Exact Solution*)

Dalam hal ini akan ada suatu nilai penghitungan. *Heuristic* memecahkan kompleksitas ini dengan aturan pencarian hanya disekitar ruang yang sebagian besar memberikan solusi yang layak (*promising*). Sehingga algoritma *heuristic* ini dapat menaklukkan kombinatorial yang menemukan solusi yang dapat diterima.

Pada *heuristic search* didefinisikan sebuah fungsi *heuristic* $h(n)$, yang meng-*estimasi* “seberapa baiknya” pilihan pada *state* n . Fungsi ini akan menentukan kualitas setiap *state* dalam *state-space*.

Adapun karakteristik nilai untuk $h(n)$, yaitu:

- | | |
|------------------------|---|
| $h(n) \geq 0$, | Untuk semua <i>state</i> n |
| $h(n) = 0$, | Menunjukkan bahwa n adalah sebuah <i>goal state</i> |
| $h(n)$ = tak terbatas, | Menunjukkan bahwa <i>state</i> n adalah sebuah kondisi sebuah <i>goal state</i> tidak mungkin terjangkau. |

Hasil dari *heuristic search* boleh jadi kurang optimal, tetapi dapat dikatakan cukup baik selama fungsi heuristik yang digunakan juga cukup cerdas. Walaupun heuristik merupakan gabungan dari beberapa prinsip atau informasi,

namun heuristik tidak menjamin hasil semutlak algoritma konvensional, tapi menawarkan hasil yang kebanyakan cukup spesifik untuk dimanfaatkan.

Keuntungan penerapan *heuristic search* dalam pencarian solusi, yaitu (Gunawan, 2005):

1. *Heuristic search* memiliki fleksibilitas tinggi yang memungkinkan untuk digunakan pada masalah yang kompleks dan tidak terstruktur.
2. *Blind search* menjamin ditemukannya solusi optimal, tetapi kurang layak untuk digunakan dalam komputasi karena kebutuhan (memori dan waktu) yang sangat besar.
3. Metode *heuristic* lebih sederhana untuk dipahami oleh pengambil keputusan, secara khusus apabila didukung oleh *analisis kualitatif*.
4. Suatu metode *heuristic* dapat digunakan sebagai bagian dari prosedur iteratif yang tetap menjamin ditemukannya sebuah solusi optimal

Beberapa metoda yang termasuk kedalam *heuristic search*, antara lain adalah:

1. *Generate and Test* (Pembangkit dan Pengujian)

Pada metoda ini terdapat prosedur yang penting, yaitu prosedur pembangkit (membangkitkan sebuah solusi yang mungkin) dan prosedur tes (menguji solusi yang dibangkitkan). Prinsip dari metoda ini adalah penggabungan antara *depth first search* dengan pelacakan mundur (*backtracking*), yaitu bergerak mundur kebelakang menuju pada suatu keadaan awal. Nilai pengujian berupa jawaban ya atau tidak.

2. *Hill Climbing* (Pendakian Bukit)

Metoda ini didasarkan pada kedalaman level pencarian pertama (*Depth First Search*). *Heuristic* digunakan untuk meningkatkan efisiensi pencarian. Metoda ini hampir sama dengan *generate and test*, perbedaan hanya terjadi pada *feedback* dari prosedur *test* untuk pembangkitan keadaan berikutnya. *Test* yang berupa fungsi heuristik akan menunjukkan seberapa baik nilai dari terkaan yang diambil terhadap keadaan-keadaan lain yang mungkin.

Terdapat tiga kekurangan pada metoda ini, yaitu:

- a. Tidak ada jaminan akan menemukan jalan penyelesaian (solusi).
 - b. Tidak diizinkan untuk melihat satupun dari langkah sebelumnya. Oleh karena itu tidak ada jaminan menemukan solusi.
 - c. Tidak dapat membedakan antara *local* dan *global maxima*
- ## 3. *Simulated Annealing Search* (SA)

Ide dasar dari metoda ini adalah memanfaatkan analogi antara cara pendinginan dan pembekuan metal menjadi sebuah struktur kristal dengan energi yang minimal (proses penguatan) dan proses pencarian untuk *state* tujuan minimal. Metoda ini berusaha keluar dari jebakan *local minimum*. Hal ini membuat metoda ini efektif, karena proses menemukan jalur perjalanan terpendek tanpa harus membangkitkan kemungkinan-kemungkinan solusi dan bekerja pada *open-list*

4. *Branch and Bound*

Pencarian pada metoda ini adalah mempertimbangkan jalur terbaik yang ditemukan terhadap jalur terbaik sebelumnya. Bobot perjalanan untuk masing-masing *route* diselidiki, disimpan dan digunakan untuk urutan pilihan. Metoda ini adalah pencarian lengkap dan akan menemukan *route* yang paling efisien melalui ruang status. Namun peningkatan efisiensi pencarian susah untuk dilakukan.

5. *Beam Search*

Metoda ini menghindari permasalahan ledakan kombinatorial dengan hanya mengembangkan simpul paling menjanjikan pada masing-masing tingkatan. Heuristik digunakan untuk meramalkan simpul mana yang kelihatannya mendekati ke tujuan. Metoda ini dalam ukuran waktu dan ruang cukup efisien. Hanya saja tidak optimal, artinya metoda ini tidak perlu menemukan *route* yang paling efisien dalam ruang status.

6. *Genetic Algorithms*

Genetic algorithm merupakan metode pembelajaran *heuristic* yang adaptif. Karena itu, terdapat beberapa versi dari *genetic algorithm*, yang menyesuaikan dengan permasalahan yang dihadapi. *Genetic algorithm* juga merupakan algoritma yang efektif, sederhana dan relatif mudah untuk di implementasikan.

Algoritma ini di dasarkan atas mekanisme evolusi biologis. Keberagaman evolusi biologis merupakan variasi dari kromosom antar

individu organisme. Teknik pencarian yang dilakukan algoritma ini adalah atas sejumlah solusi yang mungkin lebih dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol.

7. *Best- First Search* (BsFS)

Metoda ini didasarkan pada kedalaman level pencarian pertama (*Depth First Search*). Heuristik digunakan untuk meningkatkan efisiensi pencarian. Pada tiap-tiap langkah perluasan simpul dilanjutkan dari simpul yang paling menjanjikan dibuka, dimanapun simpul itu berada. Metoda ini adalah suatu pencarian lengkap, akan tetapi tidak optimal karena tidak harus menemukan *route* yang paling efisien dalam ruang status. Menemukan *route* terbaik merupakan bagian penting berikutnya.

8. *Greedy Best- First Search*

Prinsip dari metoda ini adalah melakukan *node expansion* terhadap *node* di *fringe* yang nilai $h(n)$ -nya paling kecil. Metoda ini akan selalu memilih *node* yang kelihatannya lebih dekat ke tujuan (*goal*). Metoda ini mirip dengan *depth first search*, yaitu mengikuti jalur tunggal sampai ke tujuan. Namun akan kembali jika menemui jalan buntu. Kekurangan dari metoda ini adalah bahwa proses ini bisa tidak pernah mencapai tujuan dan hanya berisolasi pada satu potong jalur.

9. *A* Search*

Algoritma A* dikemukakan oleh Hart, Nilsson, dan Raphael pada tahun 1968. prinsip dari metode ini adalah hindari *node* yang berada di *path* yang “mahal”. Metoda ini merupakan perbaikan dari *best-first search* dengan memodifikasi fungsi heuristiknya, dan dengan meminimumkan total biaya lintasan.

10. *Ant Colony Optimization (ACO)*

Ide tentang *Ant Colony Optimization (ACO)* pertama kali dilontarkan oleh Marco Dorigo ketika sedang menjadi mahasiswa Doctoral di Milan, Italy pada tahun 1991 dengan dibantu oleh Alberto Coloni dan juga Vittorio Maniezzo. Pertama kali metoda *Ant Colony Optimization (ACO)* diterapkan untuk permasalahan *Travelling Salesman Problem (TSP)* dan *Quadratic Assignment Problem (QAP)*.

2.5 **Algoritma *Ant Colony Optimization (ACO)***

2.5.1 **Sejarah Algoritma *Ant Colony Optimization***

Pada tahun 1996, dunia AI pun ikut belajar dari semut dengan diperkenalkannya algoritma semut atau *Ant Colony Optimization (ACO)*, sebagai sebuah simulasi multi agen yang menggunakan metafora alami semut untuk menyelesaikan *problem* ruang fisik. (Wardy, 2007)

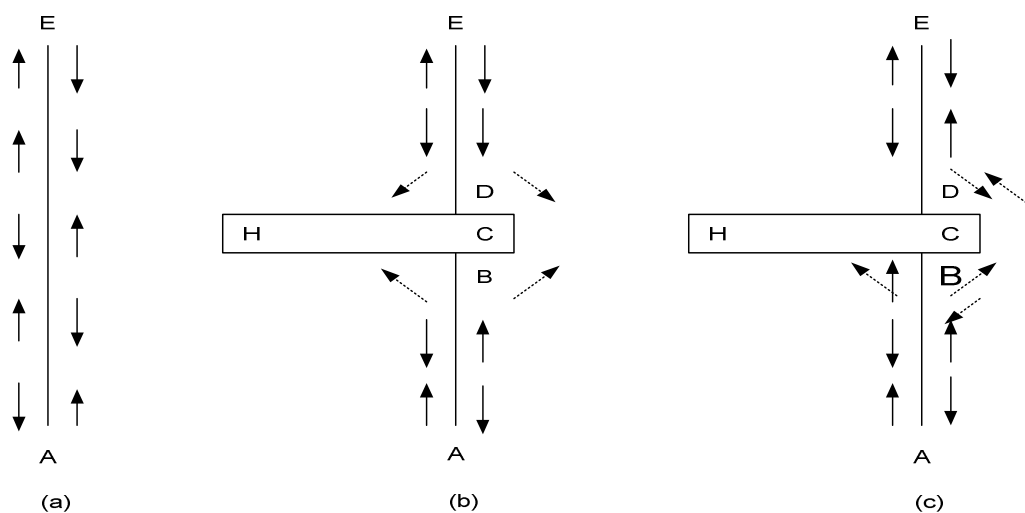
Algoritma semut diperkenalkan oleh Moyson dan Manderick dan secara meluas dikembangkan oleh Marco Dorigo, merupakan teknik probabilistik untuk menyelesaikan masalah komputasi dengan menemukan jalur terbaik melalui grafik. Algoritma ini terinspirasi oleh perilaku semut dalam menemukan jalur dari koloninya menuju makanan. (Wardy, 2007)

Pertama kali metoda *Ant Colony Optimization* (ACO) diterapkan untuk permasalahan *Travelling Salesman Problem* (TSP) dan *Quadratic Assignment Problem* (QAP).

2.5.2 Teori Algoritma *Ant Colony Optimization*

Secara alamiah koloni semut mampu menemukan rute terpendek dalam perjalanan dari sarang ke tempat-tempat sumber makanan. Koloni semut dapat menemukan rute terpendek antara sarang dan sumber makanan berdasarkan jejak kaki (feromon) pada lintasan yang telah dilalui. Semakin banyak semut yang melalui suatu lintasan, maka akan semakin jelas bekas jejak kakinya. Hal ini akan menyebabkan lintasan yang dilalui semut dalam jumlah sedikit, semakin lama akan semakin berkurang kepadatan semut yang melewatinya, atau bahkan akan tidak dilewati sama sekali. Sebaliknya lintasan yang dilalui semut dalam jumlah banyak, semakin lama akan semakin bertambah kepadatan semut yang melewatinya, atau bahkan semua semut akan melalui lintasan tersebut.

Pada gambar berikut dapat dilihat bagaimana semut menemukan jalur terpendek



Gambar 2.1 Ilustrasi Cara Semut Memilih *Path*

- 1 Semut berjalan pada jalur yang menghubungkan sumber makanan (A) dengan sarang (E). Disepanjang jalur ini terdapat jejak feromon yang dibentuk semut-semut.
- 2 Jalur yang dilalui semut dihalangai sebuah rintangan. Jadi pada posisi B, semut yang berjalan dari A ke E harus menentukan apakah ia akan bergerak ke kiri atau ke kanan. Hal yang sama pada posisi semut di D. pilihan semut untuk bergerak ke kiri atau ke kanan dipengaruhi oleh intensitas dari jejak feromon yang ditinggalkan oleh semut sebelumnya. Semakin besar jumlah feromon pada sebuah jalur, maka kemungkinan terpilihnya jalur tersebut semakin besar.

Semut pertama yang mencapai titik B atau D memiliki probabilitas yang sama untuk bergerak kekiri atau ke kanan. Hal ini dikarenakan tidak ada feromon pada jalur kiri atau kanan. Pada bagian c, dapat dilihat bahwa semut pertama yang melalui jalur BCD akan mencapai D sebelum semut pertama yang melalui BHD karena jalur BCD lebih pendek dari BHD. Semut yang berjalan dari E ke D akan menentukan jumlah feromon yang lebih banyak pada jalur DCB sehingga kemungkinan terpilihnya BCD akan lebih besar dari DHB. Lama-lama jumlah semut yang melalui BCD akan lebih banyak dari pada jumlah semut yang lewat BHD. Hal ini mengakibatkan jumlah feromon pada jalur yang lebih pendek akan bertambah secara cepat daripada jalur yang lebih panjang sehingga kemungkinan semut untuk memilih jalur lebih pendek lebih besar. Akhirnya, semua semut akan memilih jalur yang lebih pendek.

2.5.3 Konsep Dasar Algoritma Semut

Algoritma semut atau siklus semut untuk penyelesaian masalah Travelling Salesman Problem, diberikan sebagai berikut:

Terdapat N buah kota: C_1, C_2, \dots, C_N ; dengan jarak antara kota C_i dan C_j adalah $d(C_i, C_j)$. Kita diminta untuk menentukan suatu jalur p yang meminimumkan:

$$\left(\sum_{i=1}^{N-1} d(C_{\pi(i)}, C_{\pi(i+1)}) \right) + d(C_{\pi(N)}, C_{\pi(1)})$$

Kunjungan pada TSP disini akan memiliki kota asal dan kota tujuan yang sama. Misalkan jumlah semut dikota ke-i pada saat t adalah $b_i(t)$ dengan $i=1,2,\dots,N$; maka jumlah total semut pada semua kota, $m = \sum_{i=1}^N b_i(t)$.

Pada siklus semut, setiap semut akan berperan sebagai agen yang:

1. akan berpindah dari kota i ke kota j, pada interval antara t dan (t+1). Kota j dipilih berdasarkan fungsi probabilitas terhadap jarak $d(C_i, C_j)$ dan jumlah jejak yang ada pada edge yang menghubungkan antara C_i dan C_j .
2. akan mengunjungi kota yang belum pernah dikunjungi sebelumnya. Hal ini diimplementasikan dengan cara menyimpan semua kota yang telah dikunjungi semut hingga semut tersebut menyelesaikan perjalanannya, dan akan mereset kembali setelah semut tersebut menyelesaikan satu siklus
3. setelah selesai dalam satu siklus, semut tersebut akan meninggalkan jejak pada setiap edge yang dilaluinya.

Misalkan $\tau_{ij}(t)$ adalah intensitas jejak semut pada edge (i,j) pada saat t, maka setelah satu semut menyelesaikan perjalanannya dalam satu siklus, intensitas jejak akan berubah menjadi

$$\tau_{ij}(t+N) = \rho \tau_{ij}(t) + \Delta \tau_{ij}(t, t+N)$$

Dengan ρ suatu koefisien yang bernilai antara 0 sampai 1, demikian hingga $(1-\rho)$ menunjukkan penguapan jejak, dan

$$\Delta \tau_{ij}(t, t+N) = \sum_{k=1}^m \Delta \tau_{ij}^k(t, t+N)$$

Dengan $\Delta \tau_{ij}^k(t, t+N)$ adalah jejak yang ditinggalkan oleh semut k pada edge (i,j) pada antara t sampai (t+1), yang dihitung sebagai berikut:

$$\Delta \tau_{ij}^k(t, t+N) = \begin{cases} \frac{Q}{L^k}; & \text{jika semut k menggunakan edge (i, j) dalam perjalanannya} \\ 0; & \text{jika semut k tidak menggunakan edge (i, j) dalam perjalanannya} \end{cases}$$

dengan Q adalah suatu konstanta dan L^k adalah panjang jalur untuk perjalanan semut k.

Intensitas jejak semut pada saat $t=0$ untuk setiap edge (i,j) , $\tau_{ij(0)}$, diset dengan sembarang nilai yang cukup kecil. Tabu list digunakan untuk menyimpan kota-kota yang telah dikunjungi oleh semut. $\text{Tabu}_k(s)$ adalah kota ke- s yang dikunjungi oleh semut k . Karena pada TSP setiap kota hanya boleh dikunjungi satu kali, maka kota-kota yang telah tersimpan pada $\text{Tabu}_k(s)$, tidak boleh dipilih lagi oleh semut k dalam satu siklus. Apabila satu siklus telah selesai dikerjakan, maka isi Tabu_k ($k=1,2,\dots,m$) harus dikosongkan.

Visibilitas η_{ij} antara kota i dan kota j , diberikan sebesar $\eta_{ij} = \frac{1}{d(C_i, C_j)}$.

probabilitas transisi dari kota i ke kota j diberikan dengan formula sebagai berikut:

$$p_{ij}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{j \in \text{dijinkan}} [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}; & \text{Jika } j \in \text{dijinkan} \\ 0; & \text{Jika } j \notin \text{dijinkan} \end{cases} \quad \text{Persamaan 2.15}$$

Dijinkan disini mengandung pengertian $\{j; j \notin \text{Tabu}_k\}$. Parameter α dan β dan digunakan untuk mengendalikan tingkat kepentingan relatif dari jejak dan visibilitas.

2.5.4 Cara Kerja Semut Mencari Jalur Optimal

Definisi Optimasi

Optimasi ialah suatu proses untuk mencapai hasil yang ideal atau optimal (nilai efektif yang dapat dicapai).

Dalam disiplin matematika optimalisasi merujuk pada studi permasalahan yang mencoba untuk mencari nilai minimal atau maksimal dari suatu fungsi nyata. Untuk dapat mencapai nilai optimal baik minimal atau maksimal tersebut, secara sistematis dilakukan pemilihan nilai variabel integer atau nyata yang akan memberikan solusi optimal. (Wardy, 2007)

Definisi Nilai Optimal

Nilai optimal adalah nilai yang didapat dengan melalui suatu proses dan dianggap menjadi suatu solusi jawaban yang paling baik dari semua solusi yang ada.

Nilai optimal dapat dicari dengan dua cara, yaitu:

1. Cara konvensional, yaitu mencoba semua kemungkinan yang ada dengan mencatat nilai yang didapat cara ini kurang efektif, karena optimasi akan berjalan secara lambat.
2. Cara kedua adalah dengan menggunakan suatu rumus atau gambar sehingga nilai optimal dapat diperkirakan dengan cepat dan tepat

Macam-Macam Persoalan Optimisasi

Persoalan yang berkaitan dengan optimalisasi sangat kompleks dalam kehidupan sehari-hari. Nilai optimal yang didapat dalam optimalisasi dapat berupa besaran panjang, waktu, jarak, dan lain-lain. Berikut ini adalah beberapa persoalan yang memerlukan optimalisasi:

1. Menentukan lintasan terpendek dari suatu tempat ke tempat yang lain.
2. Menentukan jumlah pekerja seminimal mungkin untuk melakukan suatu proses produksi agar pengeluaran biaya pekerja dapat diminimalkan dan hasil produksi tetap maksimal.
3. Mengatur jalur kendaraan umum agar semua lokasi dapat dijangkau.
4. Mengatur *routing* jaringan kabel telepon agar biaya pemasangan kabel tidak terlalu besar dan penggunaannya tidak boros.

Selain beberapa contoh di atas, masih banyak persoalan lainnya yang terdapat di kehidupan sehari-hari.

Semut mampu mengindera lingkungannya yang kompleks untuk mencari makanan dan kemudian kembali ke sarangnya dengan meninggalkan zat feromon pada jalur-jalur yang mereka lalui.

Feromon adalah zat kimia yang berasal dari kelenjar endokrin dan digunakan oleh makhluk hidup untuk mengenali sesama jenis, individu lain, kelompok, dan untuk membantu proses reproduksi. Berbeda dengan hormon, feromon menyebar ke luar tubuh dan hanya dapat mempengaruhi dan dikenali oleh individu lain yang sejenis (satu spesies) (Wardy, 2007).

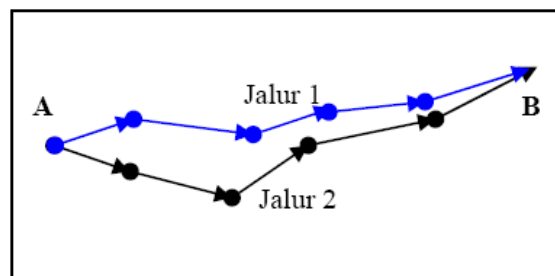
Proses peninggalan feromon ini dikenal sebagai *stigmergy*, sebuah proses memodifikasi lingkungan yang tidak hanya bertujuan untuk mengingat jalan

pulang ke sarang, tetapi juga memungkinkan para semut berkomunikasi dengan koloninya.

Seiring waktu, bagaimanapun juga jejak feromon akan menguap dan akan mengurangi kekuatan daya tariknya. Lebih lama seekor semut pulang pergi melalui jalur tersebut, lebih lama jugalah feromon menguap.

Agar semut mendapatkan jalur optimal, diperlukan beberapa proses:

1. Pada awalnya, semut berkeliling secara acak, hingga menemukan makanan.
2. Ketika menemukan makanan mereka kembali ke koloninya sambil memberikan tanda dengan jejak feromon.
3. Jika semut-semut lain menemukan jalur tersebut, mereka tidak akan bepergian dengan acak lagi, melainkan akan mengikuti jejak tersebut.
4. Kembali dan menguatkannya jika pada akhirnya mereka pun menemukan makanan.
5. Seekor semut yang secara tidak sengaja menemukan jalur optimal akan menempuh jalur ini lebih cepat dari rekan-rekannya, melakukan *round-trip* lebih sering, dan dengan sendirinya meninggalkan feromon lebih banyak dari jalur-jalur yang lebih lambat ditempuh.
6. Feromon yang berkonsentrasi tinggi pada akhirnya akan menarik semut-semut lain untuk berpindah jalur, menuju jalur paling optimal, sedangkan jalur lainnya akan ditinggalkan.
7. Pada akhirnya semua semut yang tadinya menempuh jalur yang berbeda beda akan beralih ke sebuah jalur tunggal yang ternyata paling optimal dari sarang menuju ke tempat makanan.



Gambar 2.2 Lintasan Awal Semut Menuju Tempat Makanan

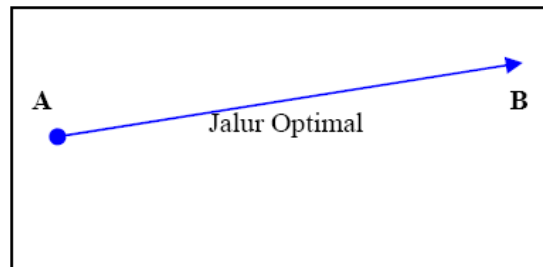
Keterangan Gambar 2.2 :

A : Tempat awal koloni (sarang)

B : Tujuan koloni semut (makanan)

Jalur 1 (biru): Lintasan yang ditempuh oleh semut 1

Jalur 2 (hitam): Lintasan yang ditempuh oleh semut 2



Gambar 2.3 Lintasan Optimal Semut Menuju Tempat Makanan

Keterangan Gambar 2.3 :

A : Tempat awal koloni (sarang)

B : Tujuan koloni semut (makanan)

Jalur Optimal : Jalur yang dilewati semut setelah beberapa iterasi

Seluruh proses ini menunjukkan berlangsungnya optimisasi alami semut yang bisa di tiru dalam kehidupan sehari-hari.

2.6 Penerapan Algoritma Semut

Algoritma optimisasi koloni semut telah digunakan untuk menghasilkan penyelesaian yang mendekati optimal. Aplikasi algoritma semut dalam kehidupan sehari-hari mencakup beberapa persoalan, yaitu:

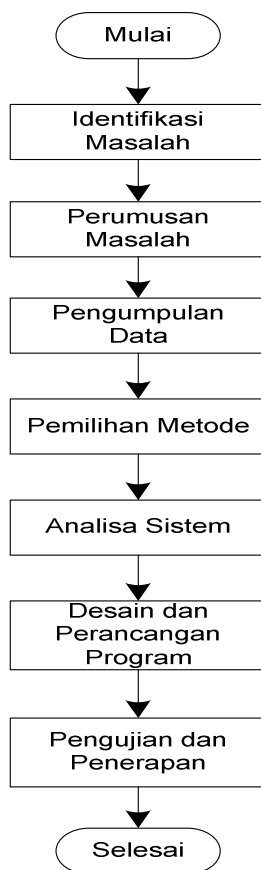
1. *Traveling Salesman Problem* (TSP), yaitu mencari jalur terpendek dalam sebuah graf menggunakan jalur hamilton.
2. *Quadratic Assignment Problem* (QAP) yang berusaha meng-assign sejumlah *n resources* untuk ditempatkan pada sejumlah *m lokasi* dengan meminimalisir biaya *assignment*.
3. *Job-shop Scheduling Problem* (JSP) juga salah satu contoh aplikasi algoritma semut untuk menjadwalkan sejumlah *j pekerjaan* menggunakan sejumlah *m mesin* demikian sehingga seluruh pekerjaan diselesaikan dalam waktu yang minimal.

4. Pengaturan jalur kendaraan
5. Pewarnaan graf
6. Network routing.

BAB III

METODOLOGI PENELITIAN

Dalam metodologi penelitian di jabarkan tahapan-tahapan yang dilakukan dalam penelitian. Metodologi penelitian terdiri dari beberapa tahapan yang saling terkait secara sistematis. Tahapan ini diperlukan untuk memudahkan dalam melakukan penelitian. Tahapan yang dilakukan dalam penelitian adalah sebagai berikut :



Gambar 3.1 Tahapan Pelaksanaan Tugas Akhir

3.1 Identifikasi Masalah

Dari latar belakang, diketahui bahwa sistem penjemputan barang di PT.TIKI masih bersifat biasa tanpa menggunakan metode yang menyebabkan terjadinya keterlambatan dalam penjemputan barang.

3.2 Perumusan Masalah

Berdasarkan identifikasi masalah yang telah dijelaskan sebelumnya, maka perlu dibuat suatu Peta Interaktif dengan pendekatan Algoritma Semut dalam pemilihan rute terpendek, sehingga tidak memerlukan waktu yang lama dan jarak yang panjang dalam penjemputan barang.

3.3 Pengumpulan Data

Pengumpulan data dilakukan dengan cara :

1. Wawancara (*Interview*)

Wawancara dilakukan dengan narasumber di PT.TIKI yaitu bidang perjalanan untuk memberi data-data lengkap tentang prosedur penjemputan barang dan pengiriman barang ke wilayah-wilayah.

2. Studi Pustaka (*Libery Research*)

Studi pustaka dilakukan untuk mendapatkan teori serta konsep yang mendukung dalam penelitian dan berkaitan dengan masalah yang diangkat dalam penelitian. Hal yang dipelajari dalam studi pustaka antara lain definisi peta, pemodelan sistem, pemrograman, dan metode yang dapat digunakan untuk kasus penjemputan barang. Teori serta konsep tersebut diperoleh dengan membaca buku-buku, jurnal-jurnal, artikel-artikel dan referensi yang terkait sehingga memudahkan dalam menyelesaikan permasalahan yang ada.

3.4 Pemilihan Metode

Metode yang digunakan untuk membangun Peta Interaktif ini adalah metode Algoritma Semut. Alasan pemilihan metode Algoritma Semut ini adalah :

Secara alamiah koloni semut mampu menemukan rute terpendek dalam perjalanan dari sarang ke tempat-tempat sumber makanan. Koloni semut dapat

menemukan rute terpendek antara sarang dan sumber makanan berdasarkan jejak kaki (feromon) pada lintasan yang telah dilalui. Semakin banyak semut yang melalui suatu lintasan, maka akan semakin jelas bekas jejak kakinya. Hal ini akan menyebabkan lintasan yang dilalui semut dalam jumlah sedikit, semakin lama akan semakin berkurang kepadatan semut yang melewatinya, atau bahkan akan tidak dilewati sama sekali. Sebaliknya lintasan yang dilalui semut dalam jumlah banyak, semakin lama akan semakin bertambah kepadatan semut yang melewatinya, atau bahkan semua semut akan melalui lintasan tersebut. Metode algoritma ini sangat cocok dengan kasus diatas yang mana dapat menemukan lintasan terpendek berdasarkan jejak kaki semut (pheromone).

3.5 Analisa Sistem

Analisa sistem dilakukan dengan dua tahap yaitu :

3.5.1 Sistem berjalan

Analisa pada sistem berjalan guna mengambil masukan yang dibutuhkan pada sistem yang akan dibangun.

3.5.2 Sistem informasi yang akan dibangun

Analisa dilakukan terhadap sistem baru berdasarkan data yang diperoleh dari perusahaan kemudian data-data tersebut digunakan dalam membangun sistem dengan menggunakan peta interaktif dan algoritma semut.

3.6 Desain dan Perancangan Program

Desain sistem informasi ini berisi Gambar, Tabel, *Flow Chart*. Setelah desain dikerjakan, desain yang dibuat dituangkan kedalam bentuk program komputer dengan menggunakan bahasa pemrograman *Delphi*. Perancangan program yang dilakukan dibuat untuk memenuhi fungsi-fungsi analisis jalur terpendek

3.7 Pengujian dan Penerapan

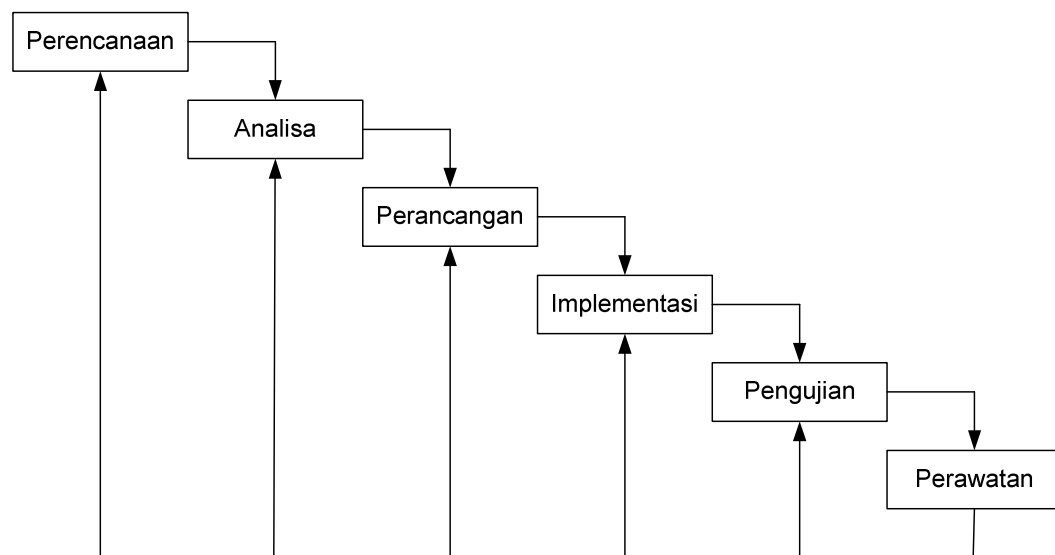
Tahap pengujian dilakukan untuk mengevaluasi hasil dari sistem yang telah dibuat, apabila sistem sudah dan sesuai dengan tujuannya maka tahap

selanjutnya akan dilakukan penerapan sistem informasi tersebut sebagai alat bantu pada perusahaan.

3.8 Metode Pengembangan Sistem

Pada dasarnya siklus hidup perangkat lunak merupakan suatu proses desain atas program yang akan dibangun untuk mendapatkan hasil yang *workable*.

Pada penelitian ini pengembangan sistem menerapkan konsep pemodelan Air Terjun (*Waterfall Model*). Berikut gambar kerangka kerja *Waterfall Model*



Gambar 3.2 Kerangka Kerja *Waterfall Model*

BAB IV

ANALISIS DAN PERANCANGAN

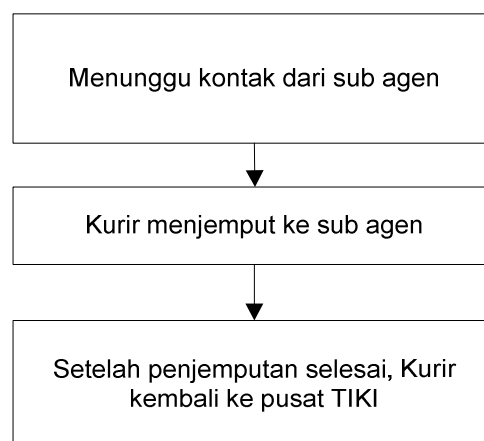
4.1 Analisis Sistem

Pada tahapan ini akan dianalisis tentang sistem yang ada dan sistem yang akan dikembangkan serta menganalisis kebutuhan sistem itu sendiri.

4.1.1 Analisis sistem yang sedang berjalan

Analisis sistem lama dilakukan untuk mendapatkan informasi penting dan menjadi sumber daya bagi sistem yang akan dikembangkan agar mampu mengatasi kelemahan-kelemahan yang ada pada sistem lama. Berikut ini adalah upaya yang dilakukan oleh PT. TIKI Pekanbaru dalam penjemputan barang.

Penjemputan barang yang dilakukan PT. TIKI adalah dengan menunggu adanya kontak dari sub agen TIKI, apakah ada barang yang akan dijemput atau tidak. Kemudian jika ada barang yang akan dijemput, PT. TIKI akan mengirimkan kurirnya untuk menjemput barang tersebut. Dalam penjemputan barang, kurir yang menjemput langsung menuju ke sub agen.



Gambar 4.1 Proses Penjemputan Barang

4.1.2 Analisis Sistem Baru

Sistem baru yang dibangun berdasarkan pengembangan dari sistem lama yaitu dengan mengumpulkan data-data penjemputan yang akan menjadi acuan dalam sistem baru.

4.2 Analisis Pengembangan Algoritma Semut Untuk Mencari Nilai Optimal

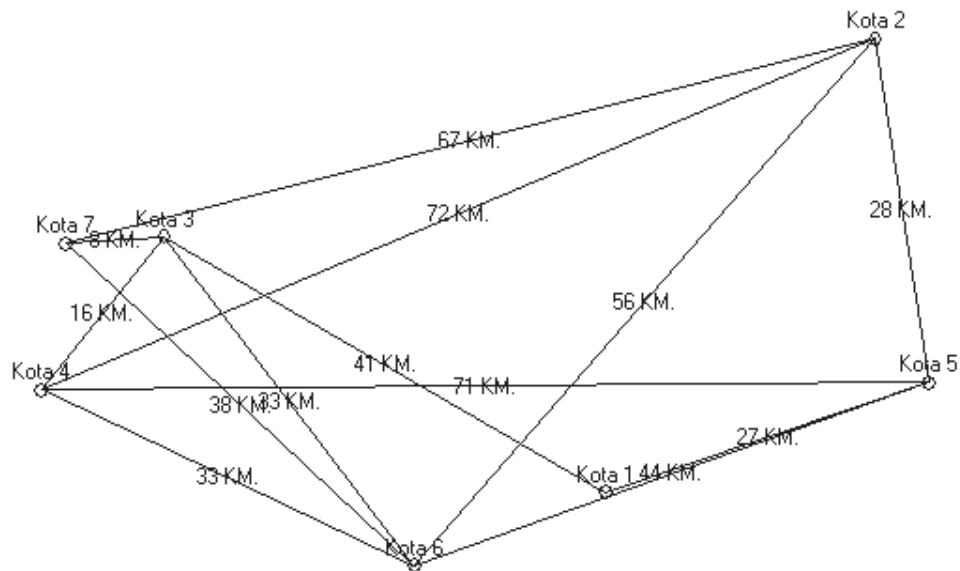
Dalam implementasi algoritma semut pada tugas akhir ini, digunakan sebuah graf yang saling terhubung antara semua node dan *bidirectional* (setiap jalur bisa ditempuh bolak-balik dua arah). Setiap busur memiliki bobot yang menunjukkan jarak antara dua buah *nodes* yang dihubungkan oleh busur tersebut.

Algoritma ini menggunakan sistem multi agen dimana seluruh koloni semut akan dikerahkan dan masing-masing bergerak sebagai agen tunggal. Setiap semut menyimpan daftar tabu list (jalan yang hanya dilewati satu kali) yang memuat *nodes* yang sudah pernah ia lalui, dimana semut tidak diijinkan untuk melalui *node* yang sama dua kali dalam satu kali perjalanan (daftar ini disebut juga sebagai jalur Hamilton, yaitu jalur pada graf dimana setiap *node* hanya dikunjungi satu kali).

Sebuah koloni semut diciptakan, dan setiap semut ditempatkan pada masing-masing *node* secara merata untuk menjamin bahwa tiap *node* memiliki peluang untuk menjadi titik awal dari jalur optimal yang dicari. Setiap semut selanjutnya harus melakukan tur semut, yaitu perjalanan mengunjungi semua *nodes* pada graf tersebut.

4.2.1. Contoh Proses Pada Algoritma Semut

Pada proses analisis algoritma semut, terdapat tujuh buah *node*/kota yang saling terhubung dengan *node* lainnya dan memiliki jarak tertentu.



Gambar 4.2 Graf Berbobot Dan Tidak Berarah Algoritma Semut

Pada dasarnya permasalahan pencarian jalur terpendek antar kota merupakan pencarian jalur terpendek antar titik yang telah diketahui koordinatnya. Dengan mengetahui konsep pencarian jalur terpendek antar titik, untuk selanjutnya dapat diterapkan pada pencarian jalur terpendek pada berbagai kota yang ingin diketahui. Contoh kasus yang akan diambil adalah pencarian jalur terpendek antara kota 1 dan kota 7.

d_{ij}	K1	k2	k3	k4	k5	k6	k7
k1	-	0	41	0	27	0	0
k2	0	-	0	72	28	56	67
k3	41	0	-	16	0	33	8
k4	0	72	16	-	71	33	0
k5	27	28	0	71	-	44	0
k6	0	56	33	33	44	-	38
k7	0	67	8	0	0	38	-

Tabel 4.1 Jarak Simpul/Node (d_{ij})

Berdasarkan contoh kasus yang ada, maka langkah yang harus dilakukan adalah:

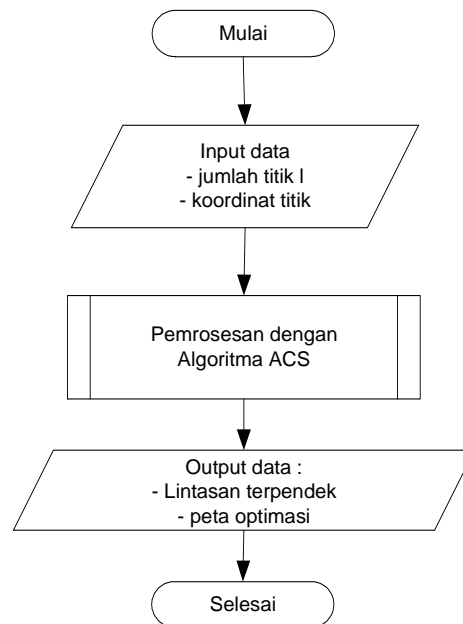
1. Menginisialisasi parameter-parameter yang diperlukan, yaitu:
 - a. τ_{ij} (intensitas jejak semut antar kota) dan perubahannya, parameter ini berfungsi menentukan jumlah intensitas jejak semut antar kota sehingga diketahui jalur terpendek yang dihasilkan.
 - b. n (banyak kota) termasuk x dan y (koordinat) atau d_{ij} (jarak antar kota), pada contoh kasus di atas, jumlah $n = 7$ dan mempunyai koordinat atau jarak yang telah ditentukan.
 - c. Q (tetapan siklus-semut), α (tetapan pengendali intensitas jejak semut), β (tetapan pengendali visibilitas), η_{ij} (visibilitas antar kota= $1/d_{ij}$), dan ρ (tetapan penguapan jejak semut), nilai dari parameter harus didefinisikan dahulu karena bersifat sebagai konstanta.
 - d. m (banyak semut), jumlah semut yang akan digunakan untuk menyelusuri jalur bisa bernilai sembarang tergantung oleh pengguna. Contoh kasus diatas menggunakan 7 semut.
 - e. NC_{max} (jumlah siklus maksimum), adalah jumlah maksimum siklus yang ingin di jalankan, hingga menemukan hasil yang terbaik. Misal maksimum siklus 2 kali, maka perhitungan akan dilakukan maksimal 2 kali hingga menemukan hasil yang terpendek.
- 2 Menempatkan kelompok semut tersebut pada kota pertama. Pemilihan kota pertama dilakukan secara acak.

- 3 Setelah menempatkan kota pertama dalam tabu list, dimulailah perjalanan semut-semut tersebut dari kota pertama menuju kota tujuan yang telah ditentukan berdasarkan persamaan probabilitas (2.1). Jarak yang dicari adalah jarak dari kota 1 ke kota 7, dengan mencari jalur terpendek dan hasil jalur yang didapatkan tidak harus melewati semua kota.
- 4 Menghitung panjang perjalanan dari masing-masing semut dan kemudian ditentukan jalur terpendek berdasarkan T_{ij} (harga intensitas jejak kaki semut).
- 5 Langkah 1 sampai langkah 4 akan diulang sebanyak NC_{max} . Setiap dimulainya siklus baru ,maka harga T_{ij} di-*reset* ulang agar bernilai sama dengan nol.

η_{ij}	1	2	3	4	5	6	7
1	-	0	0.0244	0	0.037	0	0
2	0	-	0	0.0139	0.0357	0.0179	0.0149
3	0.0244	0	-	0.0625	0	0.0303	0.125
4	0	0.0139	0.0625	-	0.0141	0.0303	0
5	0.037	0.0357	0	0.0141	-	0.0227	0
6	0	0.0179	0.0303	0.0303	0.0227	-	0.0263
7	0	0.0149	0.125	0	0	0.0263	-

Tabel 4.2 Visibilitas Antar Kota $\frac{1}{d_{ij}}$

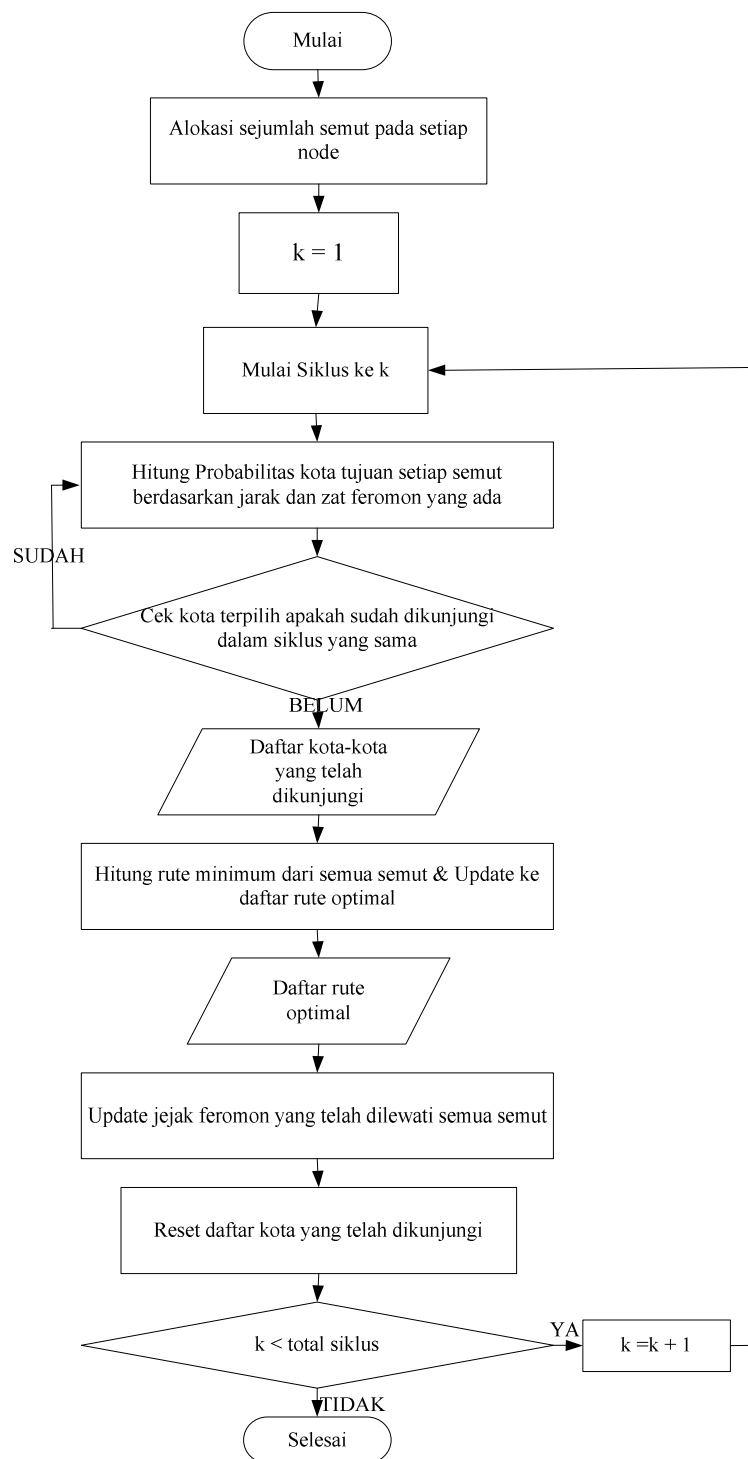
Berikut ini *flowchart* sistem pencarian lintasan terpendek dengan algoritma semut secara umum:



Gambar 4.3 *Flowchart* Algoritma ACO

- Pemilihan Rute Kunjungan

Langkah ini merupakan langkah yang digunakan semut untuk memilih kota mana yang akan ia kunjungi. Langkah ini akan digambarkan dalam bentuk flowchart pada gambar dibawah ini:



Gambar 4.4 *Flowchart* Pemilihan Rute Kunjungan

4.3 Analisis Data Sistem

4.3.1 Analisis *Input*

Komponen-komponen *input* pada sistem yang akan dibuat adalah sebagai berikut:

1. Data Penjemputan, terdiri dari:
 - a. Sub Agen TIKI
 - b. Nama Jalan
 - c. Jarak Jalan
2. Algoritma Semut, terdiri dari:
 - a. *Pheromon* awal
 - b. Perbandingan antar *random* dan perhitungan
 - c. Jumlah Semut
 - d. Faktor *update* lokal
 - e. Faktor *update* global
 - f. Faktor pembobot *pheromon*
 - g. *Loop* (perulangan)

4.3.2 Analisis Proses

Langkah-langkah yang dilakukan dalam proses penyelesaian masalah adalah sebagai berikut:

1. Cari rute terpendek dari TIKI pusat ke sub TIKI.
2. Tentukan 2 buah TIKI (A - B)
3. Tentukan jumlah simpang
4. Input jarak
5. Letak semut disimpang (X_1 X_2)
6. Pilih hitung/random
7. Dapat hasil jarak terpendek antara TIKI (A – B)
8. Lakukan hal yang sama kesemua TIKI sehingga mendapatkan jarak terpendek.

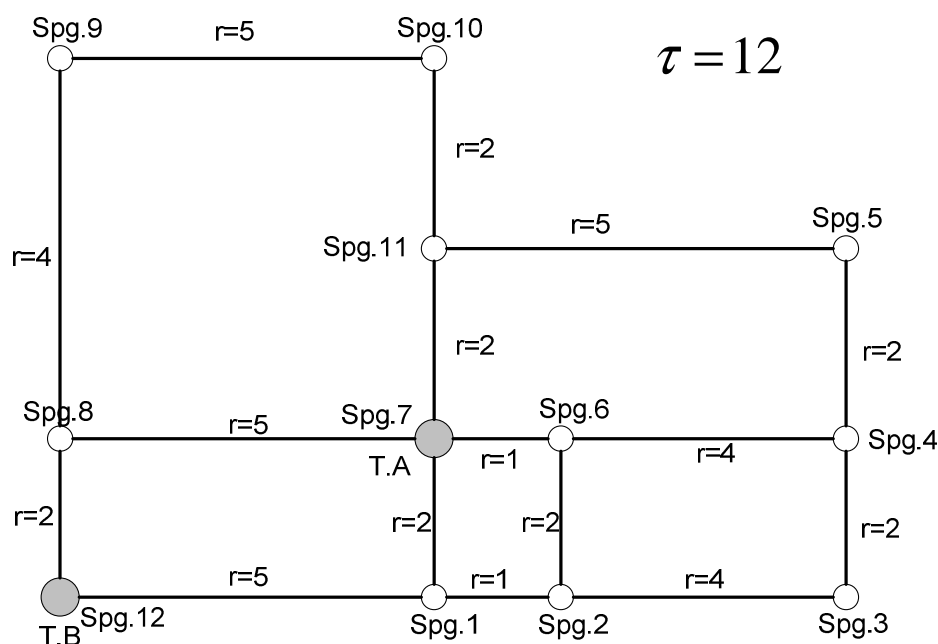
4.3.3 Analisis Output

Hasil *output* berupa tampilan peta jalur penjemputan barang dimulai dari pusat TIKI menuju sub agen yang tersebar dengan penunjukan jalur terpendek sampai kembali lagi kepusat TIKI.

4.4 Contoh Penerapan Algoritma Semut Dalam Proses Pencarian Jalur Terpendek

Dibawah ini diuraikan contoh sederhana algoritma semut untuk penyelesaian kasus pencarian jalur terpendek antara 2 buah tiki

1. Penentuan parameter *input*
 - a. 2 buah TIKI. (TIKI A – B)
2. Inputkan semua jarak antar simpang, set *pheromon* awal, misal : dgn nilai $\tau = 12$, $\alpha = 1$, $\beta = 5$



Gambar 4.5 Input jarak dan nilai pheromon

3. Letakkan semut secara acak di simpang. Misal : Semut 1 di simpang 6 dan semut 2 di simpang 8

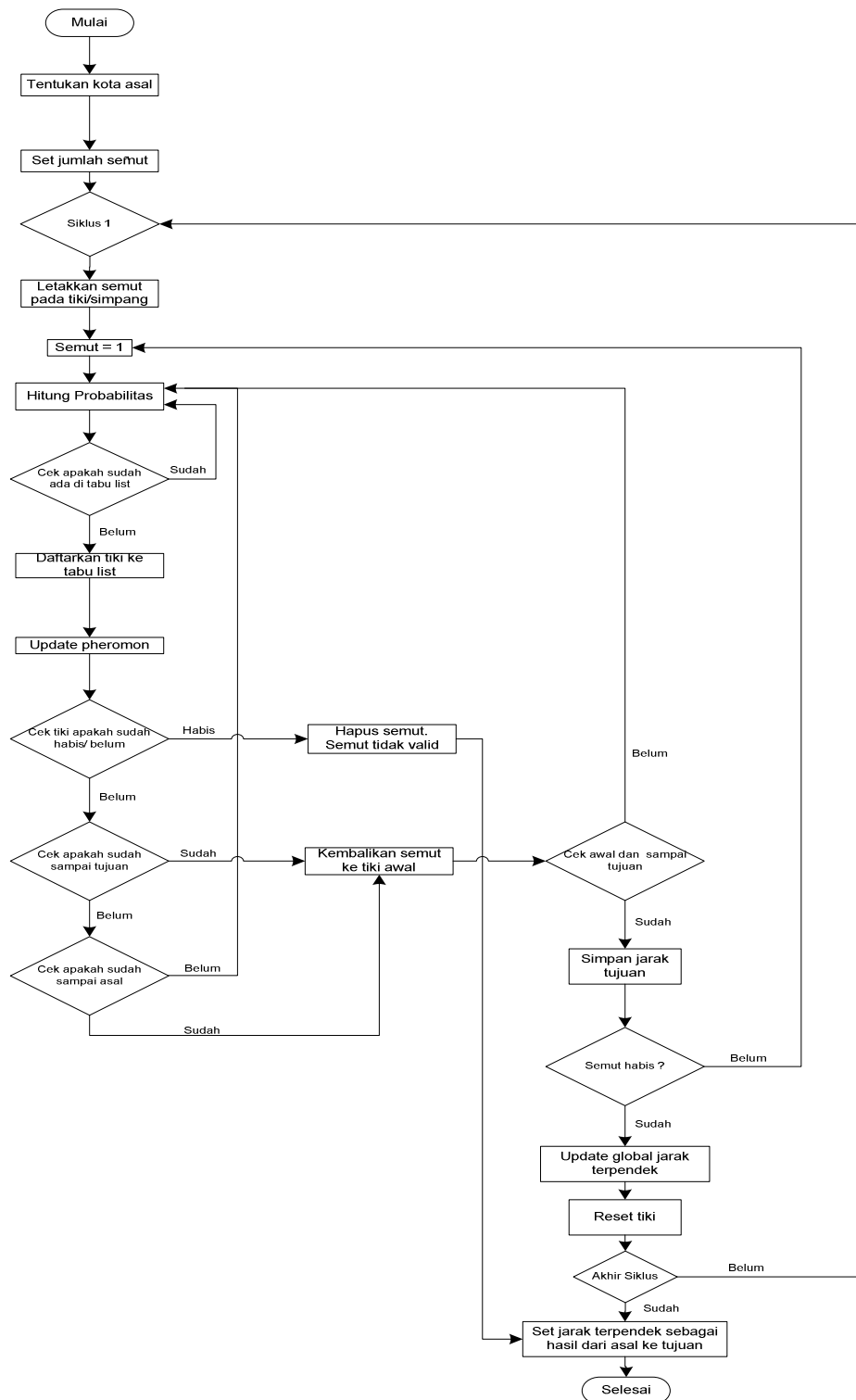
$$\begin{aligned}
p(Spg_6 - Spg_4) &= \frac{\tau(Spg_6 - Spg_4) \cdot \left(\frac{1}{rSpg_6 - Spg_4}\right)^5}{\tau(Spg_6 - Spg_2) \cdot \left(\frac{1}{rSpg_6 - Spg_2}\right)^5 + \tau(Spg_6 - Spg_4) \cdot \left(\frac{1}{rSpg_6 - Spg_4}\right)^5 + \tau(Spg_6 - Spg_7) \cdot \left(\frac{1}{rSpg_6 - Spg_7}\right)^5} \\
&= \frac{12\left(\frac{1}{4}\right)^5}{12\left(\frac{1}{2}\right)^5 + 12\left(\frac{1}{4}\right)^5 + 12\left(\frac{1}{1}\right)^5} \\
&= 0,0009
\end{aligned}$$

$$\begin{aligned}
p(Spg_6 - Spg_7) &= \frac{\tau(Spg_6 - Spg_7) \cdot \left(\frac{1}{rSpg_6 - Spg_7}\right)^5}{\tau(Spg_6 - Spg_7) \cdot \left(\frac{1}{rSpg_6 - Spg_7}\right)^5 + \tau(Spg_6 - Spg_4) \cdot \left(\frac{1}{rSpg_6 - Spg_4}\right)^5 + \tau(Spg_6 - Spg_7) \cdot \left(\frac{1}{rSpg_6 - Spg_7}\right)^5} \\
&= \frac{12\left(\frac{1}{1}\right)^5}{12\left(\frac{1}{2}\right)^5 + 12\left(\frac{1}{4}\right)^5 + 12\left(\frac{1}{1}\right)^5} \\
&= 0.968
\end{aligned}$$

7. Cari nilai p yang terbesar, tentukan sebagai simpang berikutnya, yaitu Spg
4
8. Tambahkan Spg4 ke tabu list
9. Update pheromone untuk Spg6 – Spg4
10. Jika tercapai A atau B kembalikan ke simpang awal
11. Ulangi perhitungan langkah 6 dengan memperhatikan daftar tabu list.
12. Langkah 6 sampai dengan 11 diulang sampai A dan B tercapai.
13. Hitung total lintasan semut 1 dari A-B
14. Lakukan langkah 4 untuk semut 2 sampai semut ke n
15. Pilih semut dengan lintasan paling pendek
16. Ubah nilai τ untuk masing-masing nilai τ yang dilalui semut dengan jalur terpendek dengan perhitungan
17. Lakukan langkah 1-16 sebanyak jumlah perulangan.

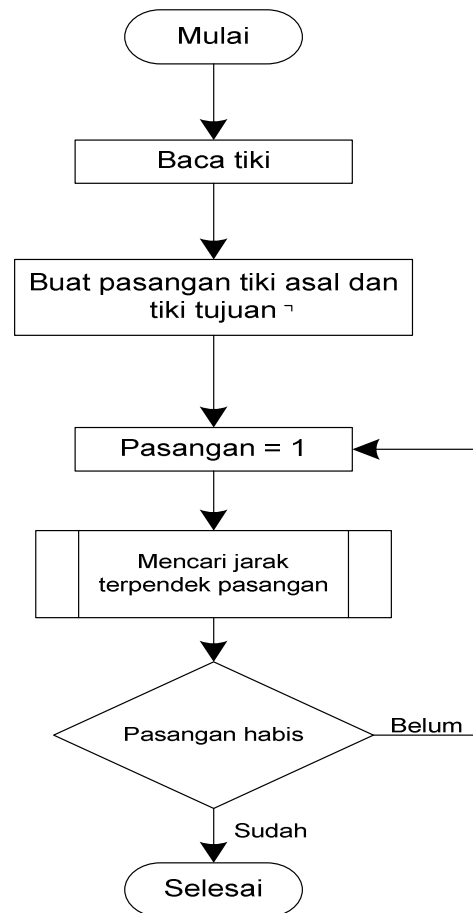
18. Semut yang melalui lintasan terpendek adalah lintasan terpendek dengan nilai optimal

4.4.1 Flowchart mencari jarak terpendek antar 2 TIKI (pasangan)



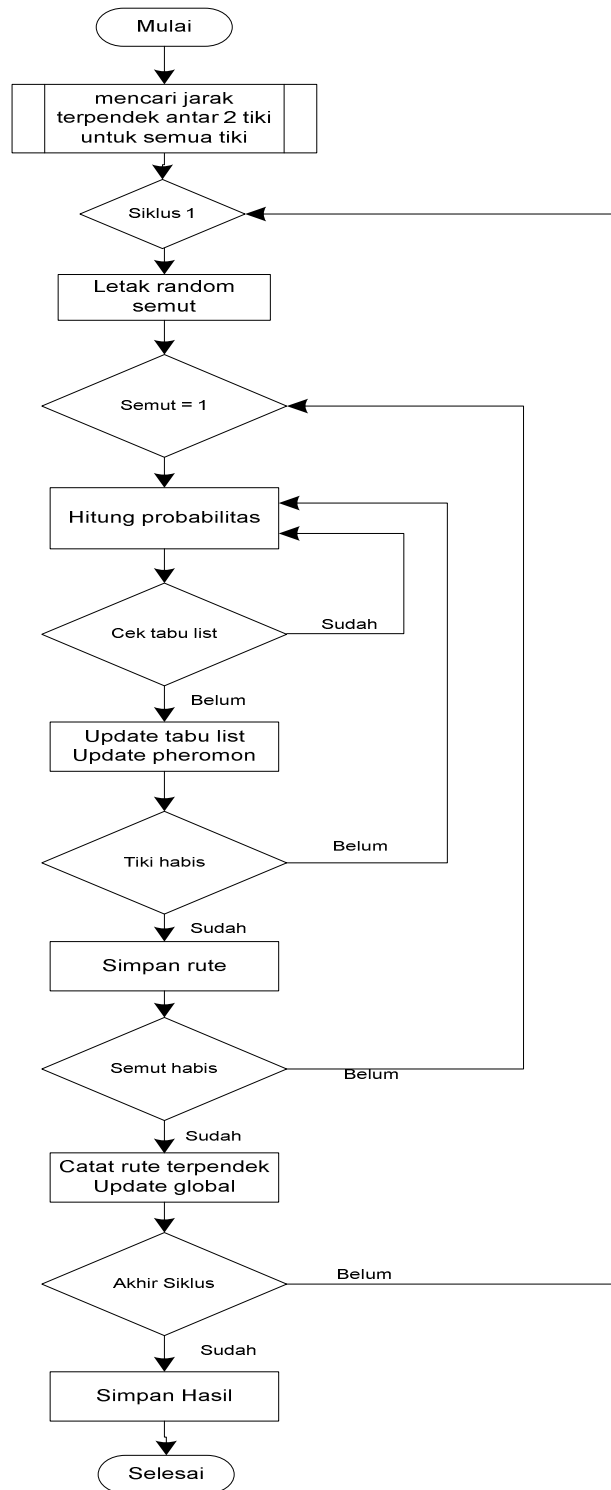
Gambar 4.7 Flowchart mencari jarak terpendek antar 2 TIKI (pasangan)

4.4.2 Flowchart mencari jarak terpendek antar 2 TIKI untuk semua TIKI



Gambar 4.8 Flowchart mencari jarak terpendek antar 2 TIKI untuk semua TIKI

4.4.3 Flowchart mencari lintasan seluruh TIKI



Gambar 4.9 Flowchart mencari lintasan seluruh TIKI

4.5 Perancangan Perangkat Lunak

Perancangan perangkat lunak pencarian jalur terpendek untuk penjemputan barang menggunakan algoritma semut terdiri dari lingkungan perancangan dan perancangan antar muka.

4.5.1 Lingkungan Perancangan

Lingkungan perancangan yang digunakan adalah sebagai berikut:

1. Perangkat Keras

- a. Processor : Intel Pentium IV
- b. Memori : 1 GB DDR2
- c. Hard disk : 80 GB

2. Perangkat Lunak

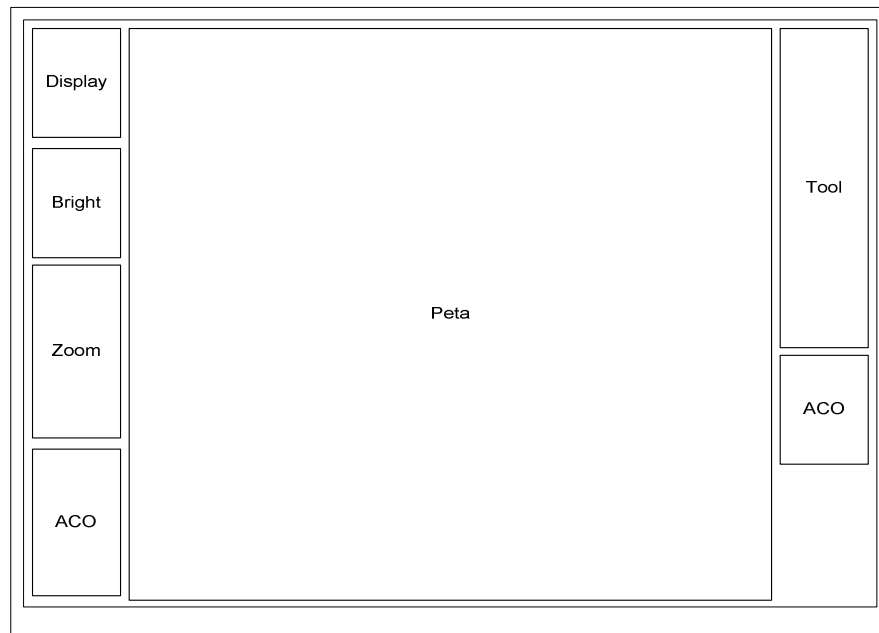
- a. Sistem Operasi : Windows XP Profesional
- b. Bahasa Pemograman : Delphi 7

4.5.2 Perancangan Antar Muka

Agar pengguna lebih mudah dalam menggunakan aplikasi yang akan dirancang, penulis memilih untuk menggunakan bahasa pemograman berbasis Windows dengan tampilan grafis dan visual yang mudah mengerti. Sehingga pengguna awam pun bisa melakukan dan mengakses aplikasi ini. Untuk kebutuhan itu penulis menggunakan bahasa pemrograman Delphi 7. Pada bagian ini, yang ditampilkan hanya perancangan menu utama, sedangkan untuk perancangan sub menu lainnya dapat dilihat pada *lampiran A*.

4.5.2.1 Form Utama

Form Utama adalah tampilan pertama yang muncul saat program dijalankan.

Gambar 4.10 *Form Utama***Tabel 4.3 Spesifikasi *Function Key* / Form Utama**

Nama Objek	Jenis	Keterangan
<i>Display</i>	Label	Mengatur tampilan
<i>Bright</i>	Label	Mengatur kecerahan
<i>Zoom</i>	Label	Mengatur besar kecil peta
<i>ACO</i>	Label	Set parameter dan keluaran
<i>Tool</i>	Label	Pengaturan file
<i>ACO Skala</i>	Label	Mengatur skala

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi Sistem

Implementasi merupakan tahap dimana sistem siap untuk dioperasikan, hal ini dilakukan setelah perangkat lunak selesai dikerjakan. Pada tahap implementasi sistem ini, diharapkan sistem yang telah dirancang siap untuk dioperasikan pada keadaan yang sebenarnya, sehingga akan diketahui apakah sistem yang dibuat benar-benar dapat menghasilkan tujuan yang diinginkan dan sesuai dengan tujuan yang diharapkan.

5.1.1 Lingkungan Implementasi

Lingkungan Implementasi sistem ada 2 yaitu: lingkungan perangkat keras dan lingkungan perangkat lunak.

1. Perangkat Keras Komputer

Perangkat keras komputer yang digunakan mempunyai spesifikasi sebagai berikut:

- a. *Processor* Pentium IV
- b. *Memory* 1 GB
- c. *Hard disk* berkapasitas 80 GB
- d. *Monitor*
- e. CD-ROM
- f. Mouse
- g. Keyboard

2. Perangkat Lunak

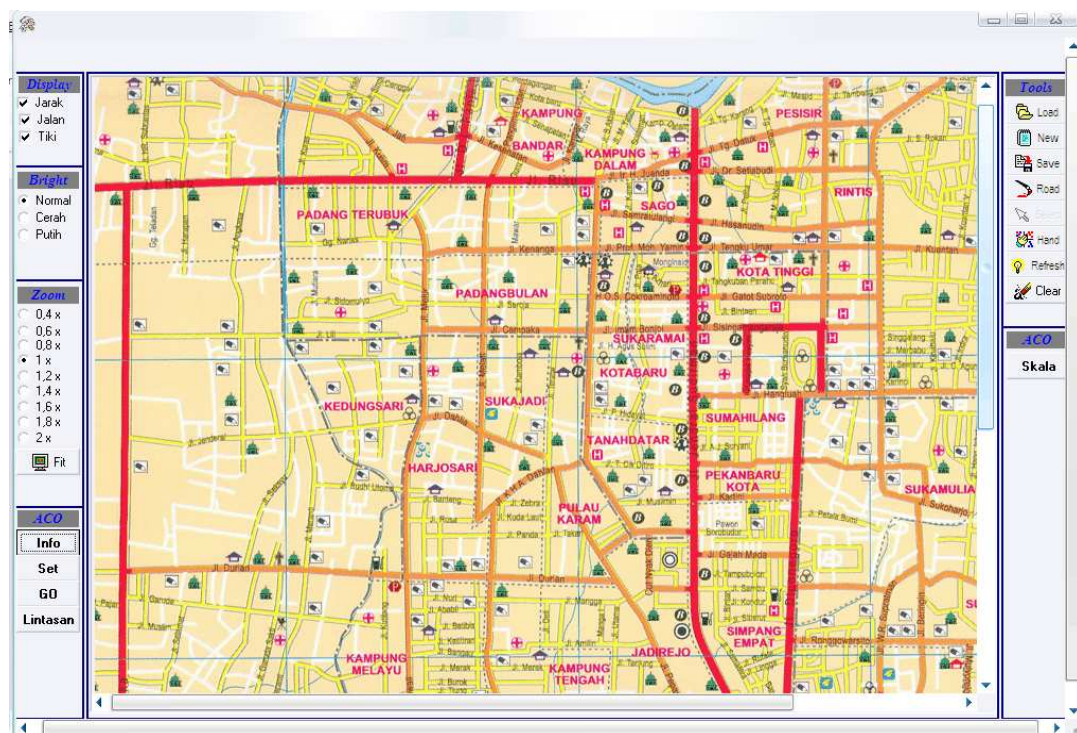
Perangkat lunak dalam implementasi ini menggunakan:

- a. Sistem Operasi Windows XP.

b. Borland Delphi 7

5.1.2 Implementasi Peta Interaktif Pencarian Jalur Terpendek Dengan Menggunakan Algoritma Semut

5.1.2.1. Menu Utama



Gambar 5.1 Tampilan Menu Utama

Ada beberapa sub menu yang terdapat pada menu utama, diantaranya:

1. Menu Display

Digunakan untuk melihat jarak, jalan dan TIKI.

2. Menu Bright

Digunakan untuk melihat lintasan dalam dan objek tiki secara detil.

3. Menu Zoom

Digunakan untuk melihat peta dengan jelas menggunakan zoom.

4. Menu ACO

- a. Info : Menampilkan lintasan jalan yang dilewati dan total jarak yang dilewati, sub TIKI yang dilewati.
- b. Set : Mengeset parameter algoritma
- c. Go : Menjalankan program
- d. Lintasan : Menampilkan lintasan peta.

5. Menu Tool

Digunakan untuk membuat lintasan, sub TIKI, nama jalan, dll.

- a. Load
Menu *load* digunakan untuk mengambil data peta dari Harddisk
- b. New
Menu *New* digunakan untuk membuat peta baru
- c. Save
Menu *Save* digunakan untuk menyimpan data peta yang telah diinput kedalam harddisk
- d. Select
Menu *Select* digunakan untuk menseleksi objek pada peta.
- e. Hand
Menu *Hand* digunakan untuk menggeser peta dalam keadaan zoom dengan tangan.
- f. Refresh
Menu *Refresh* digunakan untuk menyegarkan tampilan peta
- g. Clear
Menu *Clear* digunakan untuk membatalkan objek yang baru saja dibuat.

6. Menu Skala.

Digunakan untuk menginputkan skala peta.

Implementasi secara rinci dapat dilihat pada *lampiran B*.

5.2 Pengujian Sistem

Setelah tahap implementasi dilakukan maka dilanjutkan dengan pengujian dari implementasi yang telah dibuat. Tahap pengujian diperlukan agar dapat diketahui hasil dari program implementasi sistem.

5.2.1 Lingkungan Pengujian Sistem

Pengujian sistem ini dilakukan pada lingkungan perangkat lunak dan perangkat keras sesuai dengan lingkungan implementasi.

5.2.2 Pengujian

1. Pengujian Sistem Informasi Geografis Pencarian Jalur Terpendek dengan menggunakan Algoritma Semut, mengambil sampel beberapa TIKI. Yaitu dengan sampel 5 sub TIKI yang tersebar di Kota Pekanbaru.
1. Parameter Algoritma Semut dipilih secara hitung, diantaranya:
 - a. Jumlah Siklus = 60
 - b. Jumlah Semut = 92
 - c. Nilai Alfa = 1
 - d. Nilai Beta = 5
 - e. Nilai Q0 = 0,95
 - f. Nilai Phi = 0,80
 - g. Nilai Tho = 12
 - h. Parameter Global = 0,80

Pengujian secara rinci dapat dilihat pada *lampiran C*

5.2.3 Hasil Pengujian

Dari pengujian sistem yang telah dilakukan, sistem dapat memberikan alternatif jarak terpendek menuju objek tujuan tiki.

Hasil pengujian dapat dilihat pada *lampiran C*

5.2.4 Kesimpulan Pengujian

Berdasarkan hasil pengujian sistem dalam pencarian jalur terpendek menggunakan metode algoritma semut dengan parameter inputan pada pengujian, dapat disimpulkan sebagai berikut :

1. Solusi perjalanan terpendek yang dihasilkan dengan menggunakan Algoritma Semut akan semakin mendekati jarak yang optimum dengan semakin banyaknya semut.
2. Solusi yang dihasilkan tergantung pada pengaturan parameter inputan pada algoritma semut.

BAB VI

PENUTUP

6.1 Kesimpulan

Kesimpulan dari penelitian tugas akhir ini adalah sebagai berikut:

1. Algoritma Semut dapat digunakan untuk pencarian jalur terpendek untuk penjemputan barang pada PT.TIKI.
2. Dapat ditampilkan pada Peta Interaktif

6.2 Saran

Beberapa hal yang dapat diungkapkan sebagai saran untuk perbaikan di masa yang akan datang mengenai *Peta Alternatif Pencarian Jalur Terpendek* dengan menggunakan metode *Algoritma Semut* ini adalah sebagai berikut:

1. Mengembangkan sistem dengan mempertimbangkan faktor waktu tempuh perjalanan, biaya perjalanan, faktor kondisi jalan, atau faktor-faktor lainnya.
2. Wilayah penjemputan yang lebih besar

DAFTAR PUSTAKA

- Fitri,Muharromah, “*Analisis Algoritma Ant Colony Optimization (Aco) Untuk Permasalahan Pedagang Keliling / Travelling Salesman Problem (Tsp)*”, Tugas Akhir, UIN SUSKA RIAU, Pekanbaru, 2009.
- Gunawan, “*Heuristic Search*”, <http://www.hansmichael.com/download/Heuristics.pdf>, 2005.
- Internet, <http://id.wikipedia.org/wiki/Algoritma> pada tanggal 20 Mei 2009 Jam 21.00 Wib
- Internet,[http://www.informatika.org/~rinaldi/Matdis/2006-2007/Makalah/Makalah 060793.pdf](http://www.informatika.org/~rinaldi/Matdis/2006-2007/Makalah/Makalah%20060793.pdf) pada tanggal 20 Mei 2009 Jam 21.00 Wib
- Internet, <http://wikantika.wordpress.com/2008/05/06/apa-itu-peta/>Pada tanggal 15 November 9009 Jam 22.00 wib
- <http://nationalinks.blogspot.com/search?q=Definisi+peta>
- http://organisasi.org/pengertian_peta
- Kadir,Abdul. *Dasar Pemograman Web Dinamis Menggunakan PHP*. Yogyakarta:Andi .2005.
- Kusumadewi, Sri, “*Artificial Intelligence (Teknik dan Aplikasinya)*”, halaman 23-59, Graha Ilmu, Yogyakarta, 2003.
- Santosa, Isap. *Struktur Data menggunakan Turbo Pascal 6.0*.Yogyakarta: Andi Offset.2001
- Suyanto, “*Artificial Intelligence (Searching, Reasoning, Planning & Learning)*”, halaman 11-63, Informatika, Bandung, 2007.
- Suyoto, “*Intelegensi Buatan (Teori & Pemograman)*”,halaman 109, Gava Media, Yogyakarta, 2004.
- Wardy, Ibnu Sina, “*Penggunaan Graf Dalam Algoritma Semut Untuk Melakukan Optimisasi,*”
- Yusra, “*Heuristic Search Dengan Metoda Simulated Anneling Pada Euclidean Traveling Salesman Problem (TSP)*”, halaman II-1, UIN SUSKA RIAU, Pekanbaru, 2007.